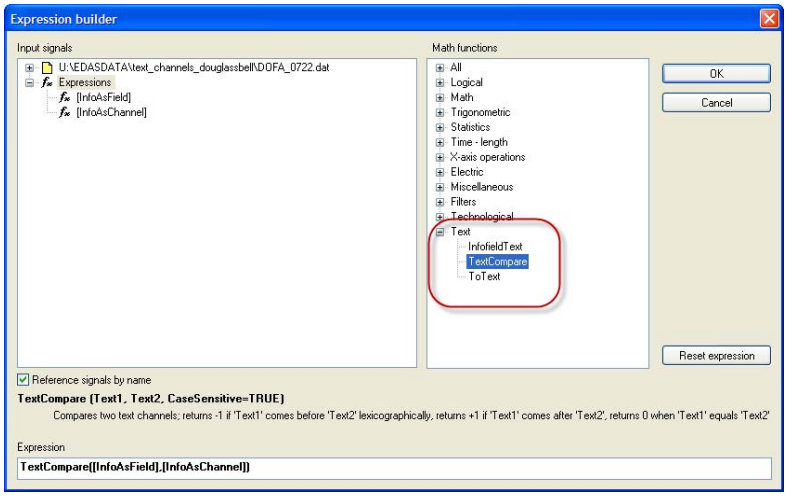
IbaAnalyzer 5.17.4 new functionality description

ibaAnalyzer text channel functions

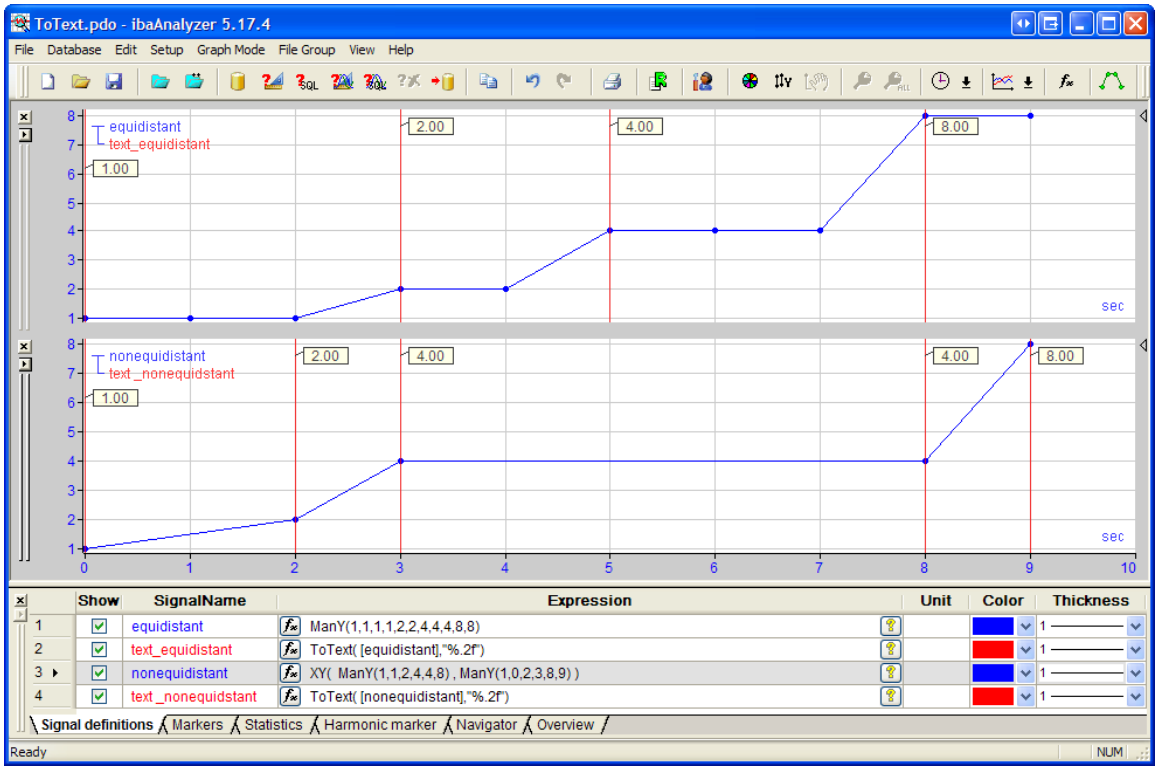
Three new functions are added to ibaAnalyzer having to do with text channels.
In the expression builder, they are available under the new function group with the header "Text".



ToText

This function converts a numeric signal to a text channel. If the input numeric signal is non equidistant, a sample is inserted in the text channel for every sample of the input signal; if the input numeric signal is equidistantly sampled, samples are inserted in the text channel only for the first sample and for every sample that has a different value than its previous sample.
This function was implemented mainly to assist in visualizing trend queries. If the numeric value were large (for instance a roll number), one had difficulty reading the exact value of the samples without switching to the marker grids.
The InfofieldText function takes the following arguments:

- Expression: The input signal to be converted.
- Format (optional): Optionally you can specify a format string that will be used to convert the floating point values of the numerical signal to a string. The format string is in C `printf` syntax. You can specify only one parameter (%) and that parameter needs to match for a 32-bit IEEE floating point. If you omit this parameter the format string "`%g`" will be used.

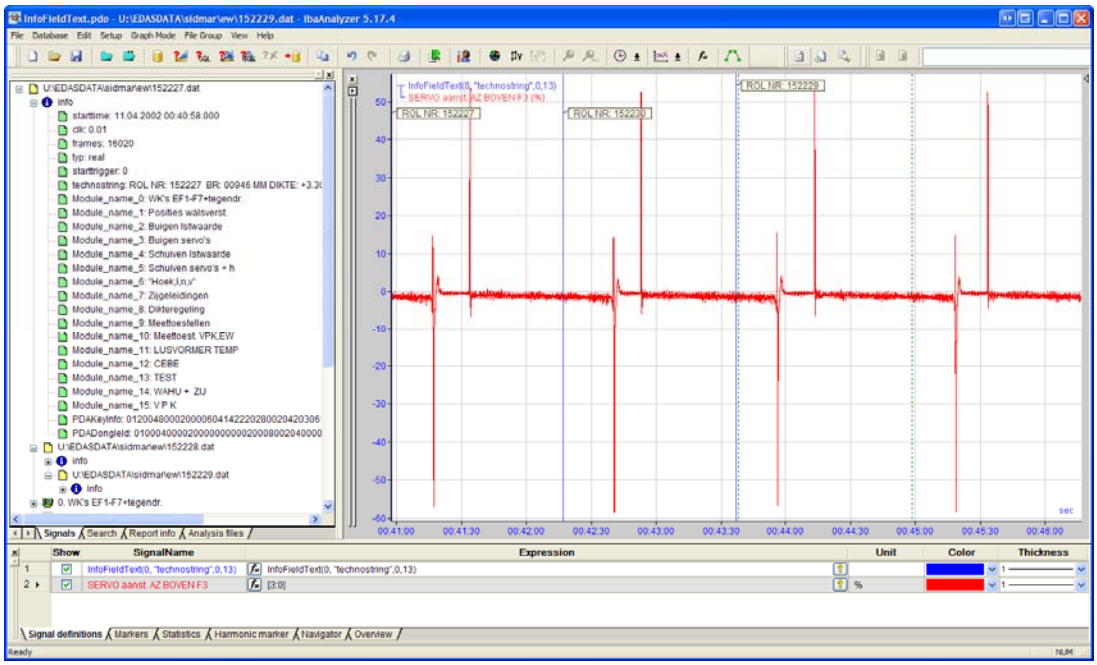


InfofieldText

This function allows you to depict an infofield or part of it as a text channel.
This function is similar as the ibaAnalyzer “Infofield” function, but instead of trying to interpret the infofield as a numerical value, this function leaves the value as text and returns a text channel. The InfofieldText function takes the following arguments:

- Fileindex: Index of the file the info field should be taken from. This index is zero based, i.e. the first loaded file is 0, the second loaded file is 1 and so on.
- Infofield: The name of the infofield. You need to use quotes (″) around this argument.
- Begin (optional): Specify here the index of the first character to be taken from the infofield. This index is zero based (i.e. starts from 0). If this and the next parameter are omitted, the entire infofield is taken.
- End (optional): Specify here the index of the last character to be taken from the infofield. This index is zero based (i.e. starts from 0). If this parameter is omitted, the infofield is taken from the character specified by the begin parameter until the end of the infofield.

If no files are appended to the file and the requested info field is present you will get a text channel with a single sample at the file start time. If the info field is not present you will get an empty text channel. If one or more files containing the info field are appended to the initial file, you will get more samples in your text channel located at either the file start times (if you have the option “synchronize on recording time” active) or located at the boundaries between the appended files (if the synchronize option is off).

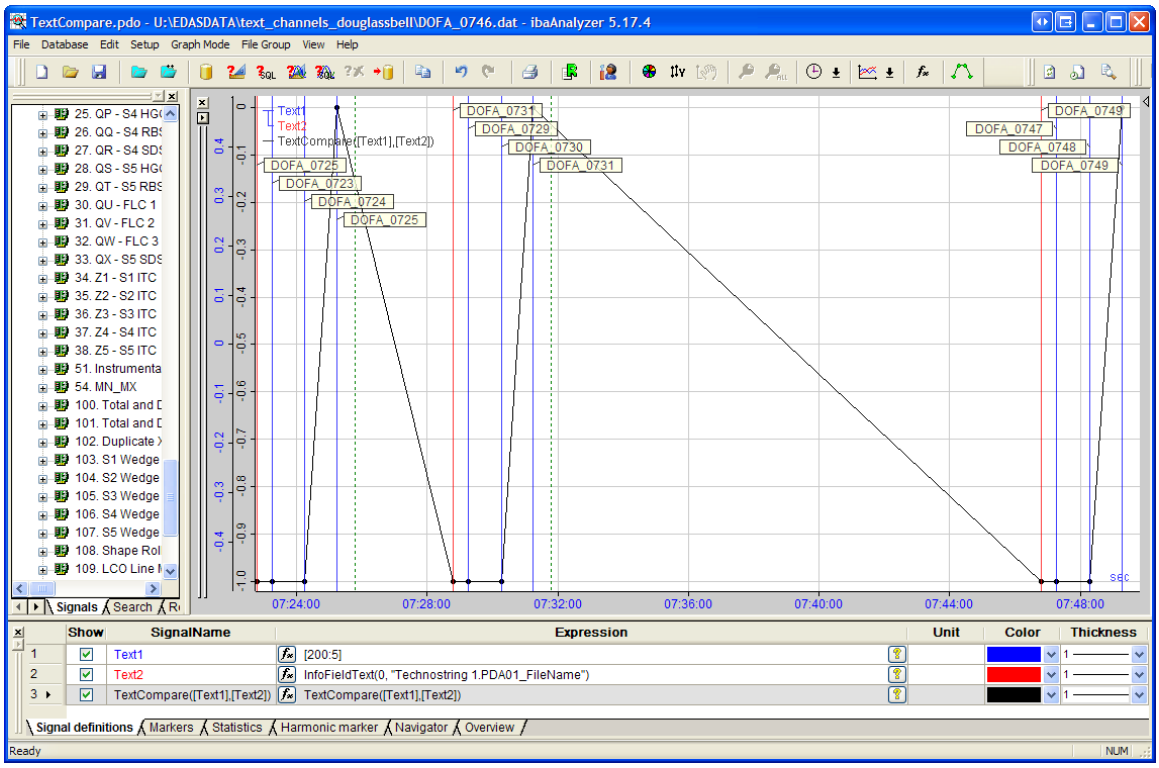


TextCompare

This function allows you two compare text data lexicographically.
The function takes the following arguments:

- Text1: The first parameter in the comparison. This can be either a text channel or a literal text. A literal text needs to be specified between quotes (″);
- Text2: The second parameter in the comparison. This can be either a text channel or a literal text.
- CaseSensitive (optional): Specify here whether or not the comparison needs to be done case sensitive. Specify TRUE if lower- or uppercase needs to be taken in account (i.e. “aaAA” and “aaaa” are considered different texts), specify FALSE otherwise. If this parameter is omitted the comparison is case sensitive.

The function returns -1 if the text of the first argument comes lexicographically before that of the second argument, +1 if it comes after and 0 if both texts are equal.
If both arguments are literal texts, a constant is returned.
If one argument is a literal text and the other argument is a text channel with more than one sample or with one sample with a timestamp different from 0, a comparison is done for each sample of the text channel and the resulting signal will have samples at the same timestamps as the text channel. If the text channel only has one sample and it is at timestamp 0, it is treated as a literal text and a constant is returned.
If both arguments are text channels, the union of the timestamps of both signals is taken and a comparison is done for each timestamp in that union. If a text channel has a sample where the other text channel has none, the text at the sample to the left is taken for the other channel. If there is no such sample the string of the present sample is compared to the empty string (the resulting sample will be 1 if the present sample is of the first channel and -1 if it is of the second channel).


TextCompare.pdo - U:\EDASDATA\text_channels_douglassbell\DOFA_0746.dat - ibaAnalyzer 5.17.4
File Database Edit Setup Graph Mode File Group View Help
Text1
Text2
TextCompare([Text1],[Text2])
DOFA_0725
DOFA_0723
DOFA_0724
DOFA_0725
DOFA_0731
DOFA_0729
DOFA_0730
DOFA_0731
DOFA_0747
DOFA_0748
DOFA_0749
DOFA_0749
07:24:00
07:28:00
07:32:00
07:36:00
07:40:00
07:44:00
07:48:00
sec
Signals Search
Show SignalName Expression Unit Color Thickness
1 Text1 [200:5]
2 Text2 InfoFieldText(0, "Technostring 1.PDA01_FileName")
3 TextCompare([Text1],[Text2]) TextCompare([Text1],[Text2])
Signal definitions Markers Statistics Harmonic marker Navigator Overview
Ready
NUM